Lecture #6

Nicholas J. Gotelli

February 04, 2016

Control Structures

If Statements

- `if` statement
- `if...else` statement
- example of code efficiency versus code readability
- `ifelse` vectorized operations
- `switch` function

For Loops

- basic structure
- use of numerical index
- exiting early
- cycling over any vector list
- some common mistakes
- for loops versus vectors

Here is some useful code from Stackoverflow:

```r
random.sample <- function(x) {
  success <- FALSE
  while (!success) {
    # do something
    i <- sample(nrow(df), 1)
    x <- df[sample(nrow(df), 1), ]
    # check for success
    success <- x$SCORE > 0
  }
  return(x)
}
```

An alternative is to use `repeat` (syntactic sugar for `while(TRUE)`) and `break`:

```r
random.sample <- function(x) {
  repeat {
    # do something
    i <- sample(nrow(df), 1)
    x <- df[sample(nrow(df), 1), ]
    # exit if the condition is met
    if (x$SCORE > 0) break
  }
  return(x)
}
```

Function	Group
runif()	1
rnorm()	2
rep()	3
seq()	4
sort()	5
order()	6
which()	1
sample()	2
ceiling()	3
floor()	4
round()	5
trunc()	6
signif()	1
sort()	2
table()	3
unique()	4
any()	5
cat()	6
range()	1
diff()	2
cumsum()	3
cumprod()	4
cummax()	5
cummin()	6